2006-03-00

Using Common Criteria Methodology to Express Informal Security Requirements

Nguyen, Thuy D.

Proc. International Symposium on Secure Software Engineering, Arlington, VA, March 2006

Proc. International Symposium on Secure Software Engineering, Arlington, VA, March 2006, pp. 75-85.

Using Common Criteria Methodology to Express Informal Security Requirements

Thuy D. Nguyen, Cynthia E. Irvine, Douglas R. Kane Jr.
Naval Postgraduate School, Monterey, California
tdnguyen@nps.edu, irvine@nps.edu, drkane@2004.usna.com

Abstract

Often, security requirements for complex systems are hard to discern because it is difficult to determine which requirements must be allocated to the system and which pertain to the system environment. In the Common Criteria framework, threat analysis results in a set of objectives that can be subdivided into two major categories: those allocated to the system itself, and the remainder to the environment. By differentiating between these two types of objectives, it is possible to avoid inappropriate requirements specification. Moving beyond systems intended to undergo evaluation, we show that the Common Criteria methodology is effective in requirements analysis for informally specified systems. As a demonstration, a worked example using a Common Criteria-based process for a requirements analysis of an on-line dissemination system is presented.

1. Introduction

When constructing a system from the ground up, developers are routinely faced with the challenge of evolving changes in the system description. Since the design process is fundamentally iterative, insights gained often feed back to the design or system functionality. In many systems, this process results in a set of informal (*ad hoc*) security requirements. To judge the soundness of these informal requirements, a coherent requirements derivation methodology is needed.

The Common Criteria (CC) is an international standard for security evaluation of IT systems [1]. The CC is often perceived as overly complex due to its large collection of interdependent functional requirements and the varying degrees of formality required for different levels of assurance. This perception has been a challenge to CC acceptance in the software engineering domain. However, recent research in applying the CC in the context of software requirements engineering suggests the usefulness of the CC within the mainstream software community [2][3].

Although the CC is not a panacea for all security engineering issues, using its paradigmatic process in an iterative approach provides a sound and practicable method for capturing and analyzing security requirements for systems that lack formal requirements specification procedures. A coherent security requirements derivation process can be devised based on the CC security environment paradigm and requirements analysis methodology. When combined with the CC requirements expression rules, while abandoning the strict use of its construct and language, this requirements derivation process provides an effective roadmap to constructive and methodical expression of informal security requirements.

The CC approach to security requirements development is to analytically derive the requirements from a set of security objectives that address all aspects of the system's security environment. The system's security environment is defined in terms of inherent threats, axiomatic security assumptions about the environment, and organizational security policies. The CC further refines the concept of security objectives into two classes: Target of Evaluation (TOE) objectives and environment objectives. (The CC refers to the system that is the subject of evaluation as the TOE). The environment objectives are often overlooked or misunderstood because the developers must infer the operational goals of the system to properly define them. For example, wide distribution of project material is a typical expectation of an open source project. However, depending on the nature of the project, the operational goals of its distribution system will differ, resulting in dissimilar security objectives for the system and its environment. Distributing project material on an unrestricted website is sufficient for an educational project, but is not

acceptable for a collaborative and, in some instances, rivalrous research project where different levels of disclosure control are needed. For the latter case, the security objective that requires the distribution system to enforce some form of user access control is commonly understood by the developers. But having system objectives alone is not enough. Security objectives should also be defined for the environment in which the distribution system operates. These will provide access control supporting mechanisms such as secure user registration and proper marking of access control attributes. Environment objectives such as these are easy to miss.

Since the security objectives are used to derive the requirements it is critical that they are correctly defined and allocated to the system or to the environment. If the security requirements are improperly declared, the end system, even if well-engineered, will not satisfy the intended operational goals.

Requirement specifications for informally defined systems often concentrate on functional requirements that describe the system's capabilities and performance. Non-functional requirements specifying the system's constraints and quality are rarely given sufficient attention, leading to difficulties in obtaining the expected level of rigor and robustness for the target system. The CC defines a structured requirements expression method that addresses this shortcoming. In the CC taxonomy security requirements are defined in terms of functional and assurance (non-functional) requirements.

This paper describes how the Common Criteria was utilized as a guiding tool for the security requirements analysis and development of the Trusted Computing Exemplar (TCX) Dissemination System. Common Criteria and TCX requirements are presented, followed by a concept design for the initial implementation of an XML-centric web-based dissemination system. A threat analysis of the proposed system, development of the system requirements, and generation of a high level design specification are included. The design specification was used to implement an initial prototype.

2. Dissemination requirements elicitation

Two sets of high level requirements drive the need for a specialized dissemination system: those of the Common Criteria and those of the Trusted Computing Exemplar (TCX) project [4].

2.1. Common Criteria requirements

For high assurance products, trusted distribution, as specified in the Trusted Computer Systems Evaluation Criteria (TCSEC) [5], mitigates two threats: tamper during Trusting Computer Base movement and distribution of a counterfeit system [6]. Technical measures and procedures must exist to assure that all customer updates distributed are identical to the master version [5].

The Common Criteria-based Separation Kernel Protection Profile (SKPP) [7] extends these requirements for a class of high assurance kernels. It outlines procedures for both the initial distribution and subsequent updates to the TOE and its components. The TOE is required to include procedures and tools to verify that the on-site version of the TOE matches the master distribution version [7].

The TCX Separation Kernel will meet trusted delivery requirements using the TCX Dissemination System.

2.2. TCX requirements

The TCX project is intended to demonstrate the application of a high assurance development methodology to the construction of products evaluatable against the highest evaluation assurance level defined by the Common Criteria. The TCX project includes four related activities: (1) creation of a reusable high assurance development framework, (2) development of a reference-implementation trusted network component, (3) support for evaluation of the reference component against the highest assurance criteria, as defined in the Common Criteria (EAL7), and (4) open dissemination of the results of the first three activities [8].

The reusable development framework and open dissemination objectives establish conditions that the TCX Dissemination System must address.

The development framework includes a high assurance rapid development environment that consists of tools and procedures including a documentation integration environment used to construct and manage the TCX project documents. XML will be used for authoring project documents. The use of XML paves the way for fine-grained access control during dissemination.

A primary objective of the TCX project is its global availability. In addition, geographically distributed project collaborators and evaluators need access to releasable project artifacts. While other open source projects make their project material available on unrestricted websites [9][10][11][12][13], the TCX project requires a controlled dissemination environment.

The open dissemination requirements for the TCX project include mechanisms for continuous contribution, evaluation and distribution of various project configuration items and deliverables [4]. The TCX Dissemination System must adhere to Common Criteria requirements for high assurance trusted delivery defined by the SKPP. The TCX Dissemination System must be web-based and able to disseminate XML documents, source code, and project artifacts in various formats with integrity and confidentiality protection. Additionally, the Dissemination System must provide group-based access control for different areas of the online site. Finally, the system must provide an easy to use interface for user access and data download.

3. Dissemination requirements analysis

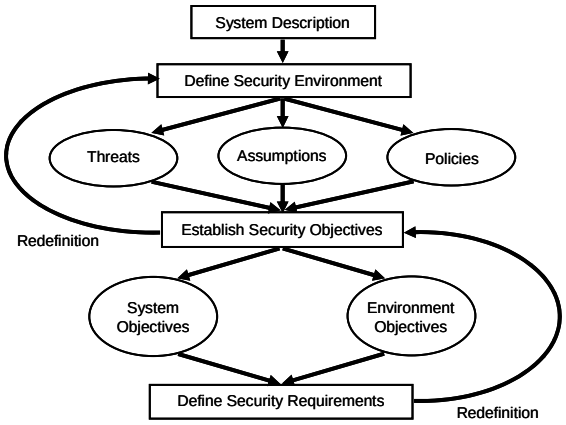

Figure 1. Requirements derivation process

The concepts and relationships of the method used to derive the security requirements for a target system are depicted in Figure 1. The requirements derivation process is a series of hierarchical refinement activities that starts with a description of the target system including its purpose, concept of operation, conceptual architecture, and system access policy. Next, the system's security environment is defined in terms of environmental assumptions, anticipated threats, and organizational security policies. Analysis of the security environment results in a set of security objectives that must be addressed by either the target system or its operational environment. The system and environment objectives together should satisfy the intended functional goals and purpose of the target system. Last, the security requirements that implement the established objectives and are levied on both the target system and its environment are developed. This process is iterative and requires the results of a given activity be traceable to the derived elements from the last higher activity. The traceability is demonstrated by the evidential material defined as rationale description in the CC mapping framework. If discrepancies exist, the results of a particular activity are invalidated and the process repeats at the previous adjacent activity.

This section describes this process in the context of the TCX Dissemination System.

3.1. High level system description

Materials from the TCX project will be disseminated on the Internet to users. The Dissemination System protects dissemination material from unauthorized access and ensures that users will receive the material with a guarantee of integrity and version, as specified by the trusted delivery requirements of the Separation Kernel Protection Profile [7]. The Dissemination System must support multiple users with varying degrees of access to materials of differing sensitivity. Thus, a single archive file distribution is not suitable.

3.1.1. Dissemination environment concept of operation. The process by which material is installed on the Dissemination System is shown in Figure 2.

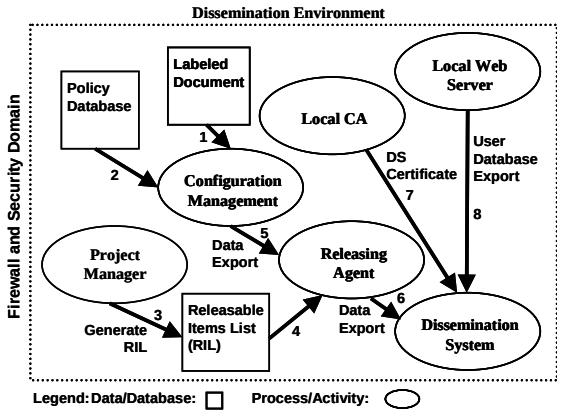

Figure 2. Dissemination environment

During development, files are labeled with sensitivity markings. In Step 1, labeled materials are submitted to Configuration Management (CM), where they are digitally signed. The TCX project will provide specific users with an external signature verification tool for checking the integrity of their downloaded material. In Step 2, the TCX Configuration Control Board submits a dissemination policy database to CM. In Step 3, the project manager generates a Releasable Items List (RIL) that specifies which documents may be released through the

Dissemination System. The RIL is passed to the Releasing Agent (Step 4). The Releasing Agent obtains the dissemination policy database and the signed materials from CM (Step 5). In Step 6, the Releasing Agent exports all releasable material, the RIL, and the policy database to the Dissemination System. The Dissemination System retrieves or updates its signed digital certificate from the CA (Step 7), and the user database (described in the next section) from the web server (Step 8). All externally generated data (i.e., Steps 6 through 8) are transferred to the Dissemination System via secure means. At this point, the Dissemination System is ready for operation.

The Dissemination System has two types of users: public users and registered users. Public users can only access non-proprietary project material. Figure 3 shows the user access flow for registered users and the retrieval of restricted project material. The user must first register with the CISR Web Server to obtain a unique user name and password before accessing dissemination material (Step 1). Once registration is complete, the user information is exported to the Dissemination System (Step 2). Finally, in Step 3 the user accesses material on the Dissemination System by logging on with the user information.

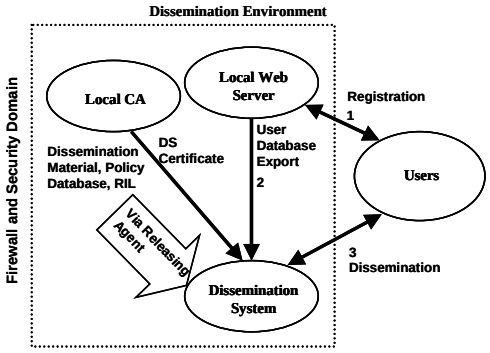

Figure 3. User data access

3.1.2. Dissemination system architecture. The Dissemination System consists of a commercial web server platform (i.e., hardware, operating system and web application server), the Dissemination Application, and various supporting tools.

The Dissemination Application (DA), on the Dissemination System, maintains the confidentiality and integrity of the dissemination data, prepares material for dissemination, and is responsible for webpage management. The DA uses access control markings on documents to generate the webpage repository. A web interface provides the only external access to TCX project materials. The Dissemination Application utilizes multiple databases (depicted in Figure 3) to perform its dissemination task.

3.1.3. System access control policy. Access to dissemination material will be based on the group to which the user belongs:

- TCX system administrators will be able to access all material necessary for the performance of their jobs.
- TCX developers will have read access to the material that they create.
- Evaluators will have read access to official material.
- External engineering collaborators will have read access to engineering releases and jointly-developed documents.
- NIST/NSA validators will have sufficient access to evaluation evidence, documents, and code necessary for validating the system evaluation.
- Customers will only have access to documents that are deemed appropriate per their use licenses.
- The general public will have the most restricted access to data.

The web application server uses access control lists to regulate the data available to different groups. Web server login is required for access to restricted material. No login is required for access to public material. Configurable audit functions will allow for auditing of user identification, accesses, and other events.

3.1.4. Document marking and viewing. XML *tags* will be used for access control markings. When material cannot be parsed as XML, for example source code and binaries, then XML document descriptor files will be used to store the markings. For each user group, the Dissemination System produces an HTML directory of the materials that members may download, based on the policy database and the document markings.

3.2. Security environment

In the CC paradigm, a threat analysis must consider the assumptions about the system and its environment, the threats to the system, and the organizational security policies that exist in the target environment. The assumptions are a means to narrow the threats and focus solely on the system being evaluated. These three elements permit development of a threat model from which requirements can be derived. In the context of this work, IT Environment describes the

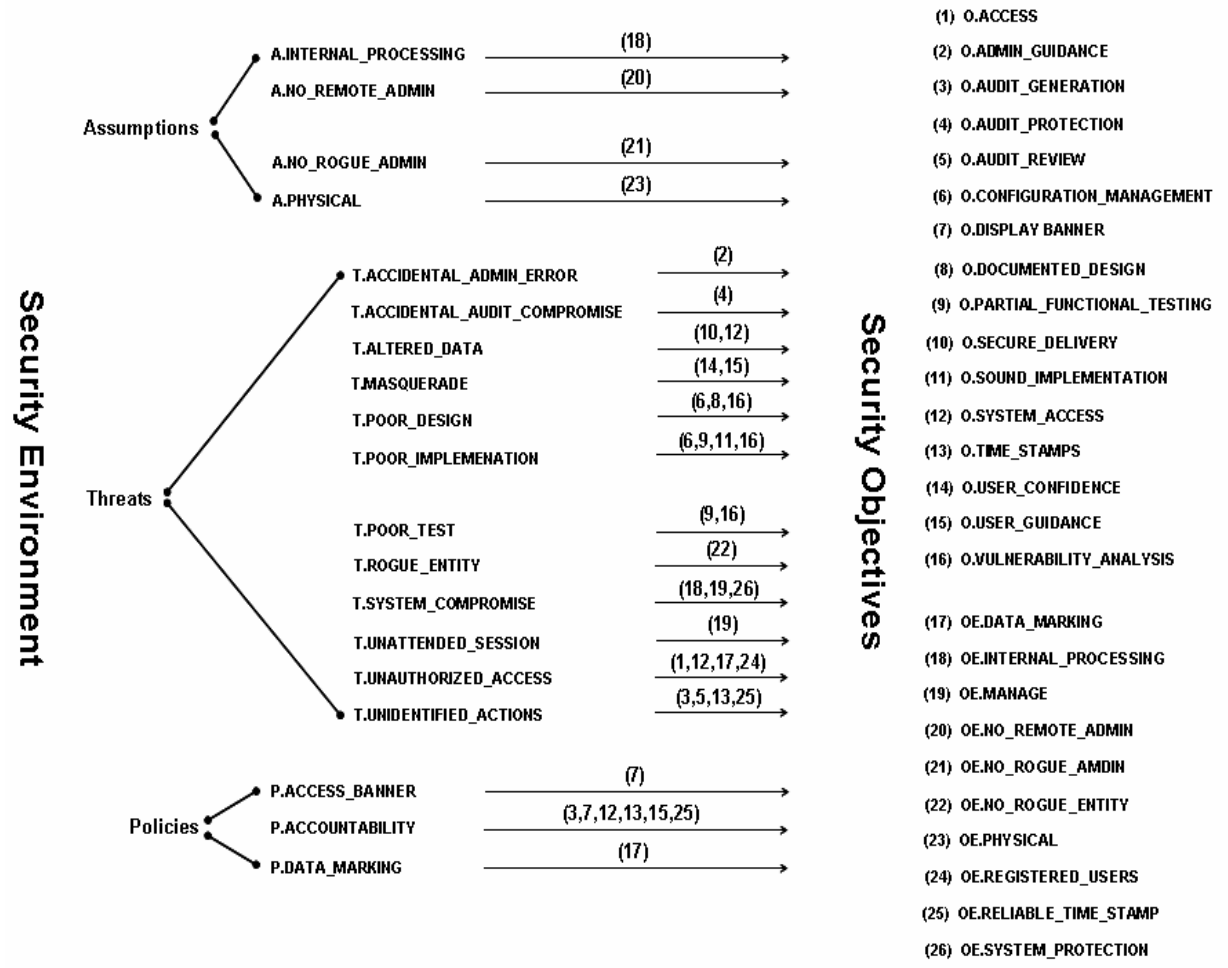

Figure 4. Security objectives mapping

underlying hardware and software upon which the Dissemination System runs. The Dissemination Environment is the Dissemination System and all of the entities described in Figures 2 and 3. Preceding threats, assumptions regarding the system must be enumerated. Threats to the Dissemination System were based on the threats to a standard web server [14] combined with additional threats resulting from the sensitivity of the information being transmitted. The high assurance requirements of the TCX project account for additional threats that would not exist in a low assurance environment.

Figure 4 shows the mapping of the assumptions, threats and policies to the security objectives defined for the Dissemination System. The TCX Dissemination System security objectives analysis [15] provides a complete description of the mapping.

3.2.1. Assumptions. Several conditions are assumed to exist in the IT environment and Dissemination Environment.

Physical security measures are assumed to be commensurate with the physical value of the data such that they will help mitigate attacks to the physical system. System administration is performed locally and by non-hostile, appropriately trained, administrators who follow all administrative guidance. Since the Dissemination System is to be hosted by a server-type platform, it is assumed that the operating system, web server, and cryptographic libraries operate correctly such that the flow of data inside the operating system, web server, and cryptographic libraries will be correct, and the implementation of this software is not flawed. System calls to the operating system will not result in incorrect Dissemination System behavior.

3.2.2. Threats. Although twelve threats have been identified [15], only the threats related to the sample

set of requirements discussed in Section 3.4 are described.

Unauthorized access is the primary concern for the system (T.UNAUTHORIZED_ACCESS). A user may gain access to dissemination data for which the user is not authorized according to the dissemination control attributes of the data.

The threat of altered data (T.ALTERED_DATA) must be mitigated. The version of dissemination data received by the end user may differ from the master version of the releasable data maintained by the Dissemination System resulting in corrupted delivery. The threat of a system compromise (T.SYSTEM_COMPROMISE) must be mitigated to ensure that the data and code of the Dissemination System is not inappropriately accessed. The realization of this threat could result in the improper dissemination of data to users. The threat of a foreign entity masquerading as the Dissemination System (T.MASQUERADE) may also result in the misrepresentation of the Dissemination System's controlled dissemination data.

The developmental threats of poor design, poor implementation and poor test are all relevant threats to a newly developed system (T.POOR_DESIGN, T.POOR_IMPLEMENTATION, T.POOR_TEST). If the design or implementation is incomplete or poor, then potential threats become a reality on the system. Unintentional errors in the requirements specification, design or implementation of the Dissemination Application may occur, leading to flaws that may be exploited by a casually mischievous user or program. Poor testing of the system, once implemented, can leave security vulnerabilities on the system undetected.

3.2.3. Organizational security policies. These policies define the security rules and acceptable use of the system within an organization, and help to maintain the integrity of the data.

The P.ACCESS_BANNER policy requires the Dissemination System present a banner to all users describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. The P.ACCOUNTABILITY policy requires that the users be held responsible for their actions while using the system. The P.DATA__MARKING policy requires that all dissemination data be properly marked with dissemination control attributes. This policy is pivotal to the secure dissemination of data.

3.3. Security objectives

To address the threats, organizational security policies and assumptions defined by the Dissemination System, a set of security objectives for both the Dissemination System and its environment are required. The Dissemination System's security objectives are developed to mitigate the threats and implement the policies of the Dissemination System. The security objectives for the environment are created to address the assumptions about the environment in which the Dissemination System operates. Only the objectives related to the sample set of requirements discussed in Section 3.4 are presented. The mapping of these objectives to the related threats/assumptions/policies is summarized in Figure 4.

3.3.1. Dissemination System security objectives. The O.ACCESS objective requires that the Dissemination System only disseminate information in accordance with the access control policy and ensure that dissemination material is properly distributed only to authorized users. The O.SECURE_DELIVERY objective requires the Dissemination System to provide mechanisms that allows the authorized users to verify that the received material is from the TCX Dissemination System and that the distributed material matches the master version maintained on the Dissemination System. The O.SYSTEM_ACCESS objective requires the Dissemination System to provide mechanisms that control a user's logical access to the system. The O.USER_GUIDANCE objective requires that all necessary information for secure data access be provided to the users. This includes guidance on the authentication methods used by the Dissemination System to authenticate itself to the clients. The O.DOCUMENTED_DESIGN objective requires that the Dissemination Application design be sufficiently and correctly documented to ensure all functional requirements are satisfied.

3.3.2. Environment security objectives. The OE.INTERNAL_PROCESSING objective requires the IT Environment upon which the Dissemination System is built to ensure that the operating system, web server, and cryptographic libraries function as expected. The OE.PHYSICAL objective requires the IT Environment to provide physical security to protect the Dissemination System and its data. These environment objectives map to assumptions and do not levy any security requirements on the IT Environment.

Conversely, the following two environment objectives impose security requirements on the IT Environment. The OE.SYSTEM_PROTECTION objective specifies that, in addition to the physical security, the IT Environment must also provide

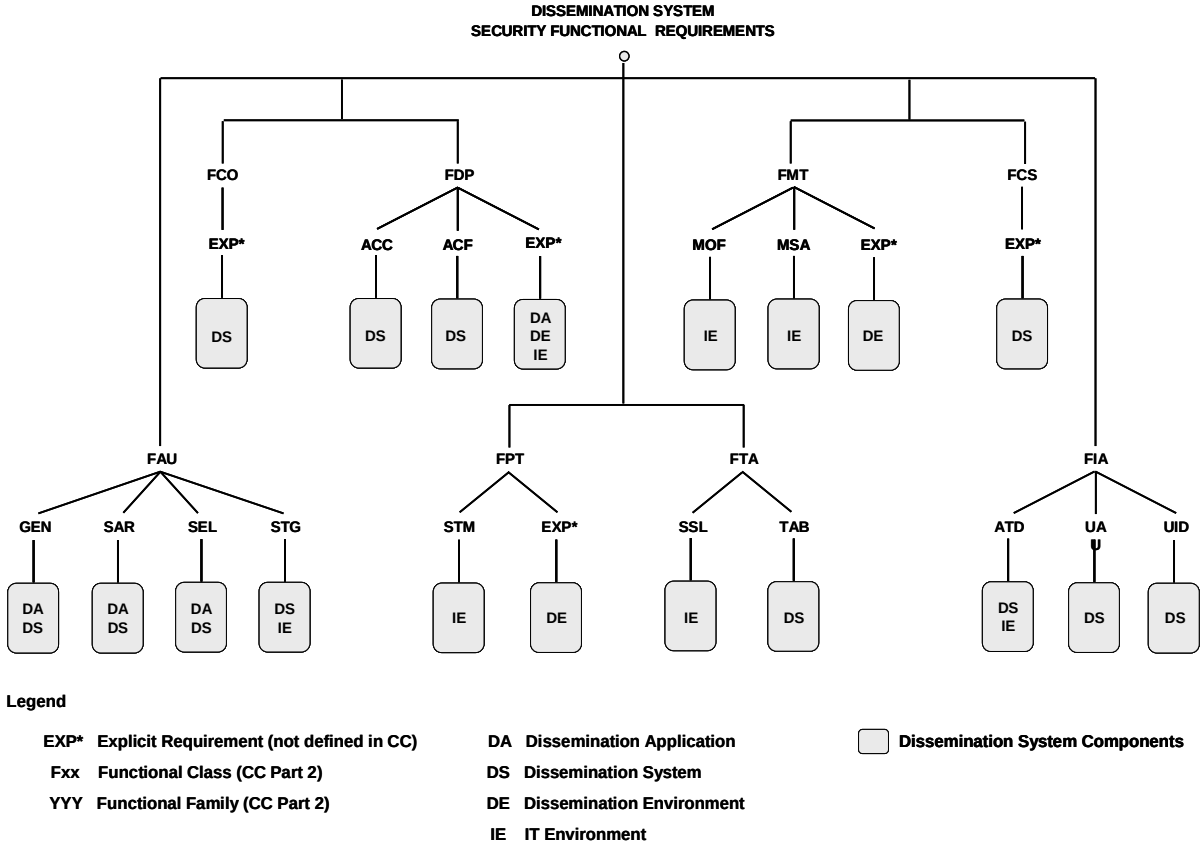

Figure 5. Security functional requirements

sufficient mechanisms to protect the data and memory on the Dissemination System during its storage and execution. The objective OE.DATA_MARKING requires that dissemination control attributes be applied to dissemination data by the appropriate entity in the Dissemination Environment. These control attributes are essential to the secure dissemination of data.

3.4. Security requirements

In the CC taxonomy, functional and assurance requirements are organized into classes, families and components based on security objectives [1]. A *class* is the most abstract expression of the approach to satisfy a set of related security objectives. A *family* within a class represents a logical grouping of requirements that collectively addresses a security problem associated with a subset of those security objectives. A *component* of a family is a set of specific security requirements for a common security need.

Although it is mandatory to use the CC-prescribed constructs for expressing security requirements, a TOE-specific extended construct can be used if the corresponding security objective(s) cannot be implemented by the predefined security requirements. Extended constructs are defined in terms of explicit requirements and the CC imposes strict rules on the expression of this type of requirements. Specifically, the explicit requirements must be modeled after the predefined requirements with respect to the level of detail, clarity and verifiability.

Navigating through the CC security requirements and their interdependencies can be daunting for software developers. But the CC framework also provides an effective way to organize the requirements for systems that are not subject to evaluation. For these systems, the predefined classes and families of functional and assurance requirements establish the structural foundation upon which informally defined security requirements can be derived and expressed. This approach was utilized to guide the security requirements engineering process for the TCX Dissemination System.

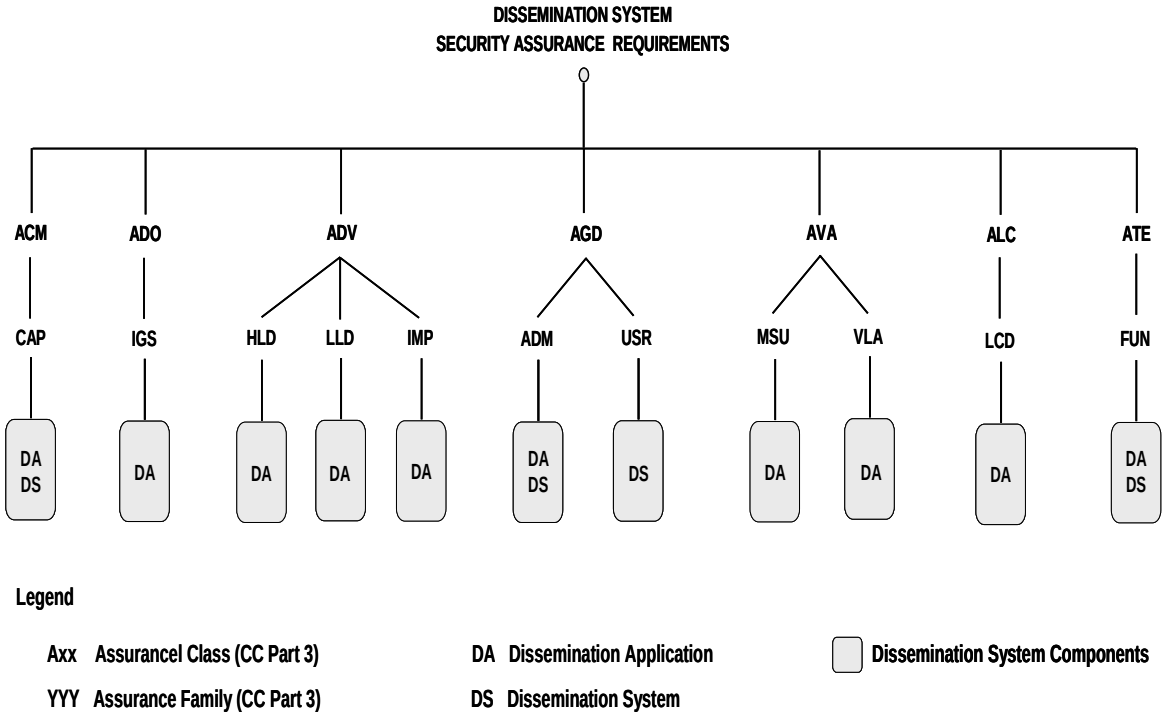

Figure 6. Security assurance requirements

Figures 5 and 6 show the CC predefined classes and families relevant to the derivation of the security requirements of the TCX Dissemination System. Leaf nodes represent the functional and assurance security requirements allocated to the different logical components of the Dissemination System. Even though the CC constructs and language were only loosely followed, the CC requirement expression methodology was useful in capturing these requirements, especially with the environmental requirements often neglected in informally defined systems.

A number of explicit requirements (shown as EXP in Figure 5) are necessary to express requirements that are unique to the Dissemination System. An example of these explicit requirements is the requirement that the Dissemination Application must redact all releasable documents. This unique requirement is represented as DA in the leaf node under the EXP family of the FDP class in Figure 5.

The CC-prescribed security requirements rationale is an effective requirements validation technique to check for improperly specified objectives or requirements or both in addition to determining the validity of the explicit requirements. During the course of the TCX Dissemination System development process, it was discovered that some of the system requirements could not meet the intended system objectives because the objectives were incorrectly defined and vice versa. Reassessment of the original set of objectives and requirements yielded a number of new environmental objectives and requirements.

As illustrated in Figure 2, the TCX dissemination environment consists of multiple entities whose security behaviors collectively contribute to the secure dissemination of TCX material. Initially, an assumption (A.NO_ROGUE_ENTITY) was made about the trustworthiness and correctness of all entities in the environment other than the Dissemination System. To satisfy this assumption, an environment objective (OE.NO_ROGUE_ENTITY) was created and additional security functional requirements were imposed on the Dissemination Application to address this environment objective. OE.NO_ROGUE_ENTITY requires the environment to ensure that entities in the dissemination environment other than the Dissemination System are trustworthy and perform their functions correctly. During the requirements review phase, it was determined that the additional functional requirements were incorrectly defined because the Dissemination Application cannot vouch for the trustworthiness and correct functionality of any entity in the operating environment. Furthermore using assumptions to address

dependencies a system has on other components in the environment in order for the system to enforce its security policy is a misuse of assumptions [16]. This triggered three changes: 1) the reclassification of the A.NO_ROGUE_ENTITY assumption into a threat (T.ROGUE_ENTITY) that entities other than the Dissemination System may not be trustworthy thus resulting in the improper dissemination of data; 2) the redefinition of the OE.NO_ROGUE_ENTITY objective; and 3) the addition of a security functional requirement on the dissemination environment. The revised OE.NO_ROGUE_ENTITY objective requires the environment to ensure that *all* entities in the dissemination environment are trustworthy and protect both the dissemination data and the data they generate to support secure distribution. The new functional requirement calls for the dissemination environment to ensure that all dissemination-related data generated by its entities is protected *in situ* and during transit.

The above example reinforces the assertion about the usefulness of the CC methodology in the analysis and expression of informal security requirements that are critical but often overlooked. The objectives/requirements traceability is further discussed in the following section.

3.5. Requirements to objectives mapping

Each objective must be traceable to one or more security requirements whose purpose is to implement that objective. However, all requirements will not map to a single objective.

Table 1 is a partial mapping of the TCX Dissemination System requirements to security objectives [15].

Table 1. Requirements/Objectives mapping

PART I: SYSTEM REQUIREMENTS MAPPING
O.ACCESS: The Dissemination System will ensure that users gain only authorized access to resources that it controls.
A.4.1 The Dissemination System shall enforce the access control policy on all registered users and data on the system. This policy shall be enforced based on the user ID and group membership for the data requested. C.2.1 The Dissemination Application shall enforce the policy based on the following dissemination file attributes: file type marking, file sensitivity marking. C.2.2 The Dissemination Application shall enforce all dissemination access control mechanisms on all dissemination material present on the server. C.2.4 The Dissemination Application shall redact each releasable document in accordance with the dissemination policy.
O.DOCUMENTED_DESIGN: The design of the Dissemination Application will be adequately and accurately documented.
D.3.1 An informal high level design specification shall be developed for the Dissemination Application. D.3.2 The design of the Dissemination Application shall meet the functional requirements.
O.SECURE_DELIVERY: The Dissemination System will provide mechanisms for users to verify that any data disseminated matches the master version maintained by the Dissemination System.
C.2.3 The Dissemination Application shall not modify the digital signatures placed on the dissemination material by the Configuration Management system.
O.SYSTEM_ACCESS: The Dissemination System will provide mechanisms that control a user's logical access to it.
A.5.1 The Dissemination System shall ensure that users are identified and authenticated in order to associate them with the proper security attributes while accessing data. Security attributes shall include but are not limited to the user's identity and the group(s) to which that user belongs. A.5.2 The Dissemination System shall authenticate registered users based on their user ID and password. A.5.3 The Dissemination System shall authenticate users prior to allowing access to any non-public documents on the Dissemination System.
O.USER_GUIDANCE: The Dissemination System will provide users with the necessary information for secure data access.
B.2.1 The user guidance shall describe the interaction between the user and the Dissemination System for proper retrieval of releasable data. B.2.2 The user guidance shall clearly present all user responsibilities necessary for secure use of the Dissemination System, including those related to assumptions regarding user behavior found in the access banner.
PART II: ENVIRONMENT REQUIREMENTS MAPPING
OE.DATA_MARKING: All dissemination data will be properly marked with access control attributes by the Dissemination Environment.
F.2.1 The Dissemination Environment shall properly mark all dissemination data with access control attributes.
OE.INTERNAL_PROCESSING: The internal implementation and execution of the Dissemination System's underlying operating system, web server, and cryptographic libraries will run as expected.
This objective addresses the assumption A.INTERNAL_PROCESSING and has no corresponding security functional requirement.
OE.PHYSICAL: Physical security, commensurate with the value of the Dissemination System and the data it contains, will be provided by the Dissemination Environment.
This objective addresses the assumption A.PHYSICAL and

has no corresponding security functional requirement.
OE.SYSTEM_PROTECTION: The IT Environment will provide sufficient mechanisms to protect the Dissemination System's data and memory during storage and execution.
E.3.1 The IT Environment shall restrict the ability to create or modify webpage content to authorized administrators. E.3.2 The IT Environment shall be capable of limiting the ability to create or modify server executable content. E.3.3 The IT Environment shall protect the Dissemination System's private key from unauthorized modification and viewing. E.4.1 The IT Environment shall restrict all non-administrative users of the Dissemination System from reading from and writing to the audit trail.

The contrast in the mapping of OE.DATA_MARKING and OE.PHYSICAL illustrates the usefulness of applying the CC methodology to requirements analysis. The former results in a functional requirement levied on the environment, whereas the latter has no functional requirement. The lack of functional requirement for OE.PHYSICAL stems from the fact that neither the Dissemination System nor the components in the environment can address the objective; instead it must be addressed through procedures. On the other hand, it is necessary to levy the stated requirement on the environment for OE.DATA_MARKING because from the requirements derivation viewpoint, security-relevant functions implemented by other components that affect the Dissemination System's ability to enforce its security policy must be considered as part of the Dissemination System. Since the access control attributes used by the Dissemination System to enforce the dissemination policy are externally applied to the data, security requirements must be levied on the environment to ensure that data marking is correctly implemented by the other components.

4. Initial implementation

The Dissemination System was implemented on a server platform running a Linux-based operating system. It consists of three major components: (1) the Web Server managed access control, audit logging, authentication, SSL protected connections, and web hosting; (2) the Dissemination Application supported audit logging, project artifact redaction, sweeping for non-releasable content, and web page management; and (3) the Administrative Tools supported audit log analysis, audit log rotation, and execution daemons.

The Apache Web Server was configured to host the test dissemination website. To enforce group-based access controls, Apache basic authentication was used. The use of the Apache rewrite engine support for runtime URL manipulation forced selected Webpage Repository directories to require SSL connections.

Administrative and supporting tools included: *logrotate*, to support audit log archiving; *webalizer*, for audit analysis; *crond*, to schedule the former two tools, OpenSSL, to generate public-private key pair and certificates; and *linkcheck.pl*, to support the elimination of broken symbolic links.

Both web server functional testing and tool testing were performed. Test scenarios exercised user authentication, group-based access controls, and HTTPS connections for non-public web content. Tools were confirmed to provide the proper Dissemination System support.

5. Future work and conclusion

Version 3 of the Common Criteria was recently released for public review [17]. The concept of IT environment has been revised and security objectives are now described in terms of the development environment and the operational environment. An interesting observation is that security requirements for the operational environment are optional in this new version. Future work on the Dissemination System includes a thorough analysis of the new requirement derivation paradigm.

Another related future work item is to enhance our development methodology with a tracking scheme to record the evolution of the objectives and requirements. In the current effort, the provenance of revisions was not captured, nor was it required by the CC. The collected statistics will provide quantifiable metrics of the efficacy of the CC Version 3 requirement derivation paradigm.

For secure software development, a holistic approach towards security requirements derivation and expression is needed. Non-functional (assurance) requirements must be considered along with functional requirements. The Common Criteria defines a security analysis methodology that is useful to express security requirements for both formally and informally defined systems. This work focuses on the latter. The CC methodology helps place limits and disciplines to the practice of developing security requirements from an informally stated system description such that the requirements can be systematically derived, represented and organized. This paper presented the TCX Dissemination System as a worked example of this proposition. Starting with a high level description of the TCX Dissemination System, a CC-based requirements derivation process was used to translate,

through analysis, the high level system requirements into a set of informal security requirements for both the Dissemination System and its environment. The CC requirements expression rules were loosely followed to help organize and represent these security requirements. A number of improperly specified objectives and requirements were discovered and redefined as the result of iteratively applying the CC traceability methodology. The worked example shows that the CC processes and rules can be soundly adapted and integrated into the requirements derivation discipline of software engineering.

Acknowledgements

We wish to thank the TCX team members. This work was sponsored in part by the Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research.

References

[1] Common Criteria Project Sponsoring Organizations, *Common Criteria for Information Technology Security Evaluation*, CCIMB-2004-01-001, Version 2.2, January 2004. http://www.commoncriteriaportal.org

[2] Yavagal, D. S., Lee, S. W., Ahn, G., Gandhi, R. A., "Common Criteria Requirements Modeling and its Uses for Quality of Information Assurance (QoIA)," *Proceedings of 43rd ACM Southeast Conference*, Atlanta, GA, March 2005.

[3] Lloyd, W. J., "A Common Criteria Based Approach for COTS Component Selection," *Journal of Object Technology*, Volume 4, Number 3, Special Issue: 6th GPCE Young Researchers Workshop 2004, pp. 25-32. http://www.jot.fm/issues/issue_2005_04/article4

[4] Irvine, C. E., Levin, T. E., Nguyen, T. D., and Dinolt, G. W., "The Trusted Computing Exemplar Project", *Proceedings of the 2004 IEEE Systems Man and Cybernetics Information Assurance Workshop*, West Point, NY, June 2004, pp. 109-115.

[5] Department of Defense, *Department of Defense Trusted Computer System Evaluation Criteria – 5200.28-STD*, 26 December 1985.

[6] Department of Defense, *A Guide to Understanding Trusted Distribution in Trusted Systems – NCSC-TG-008*, 15 December 1988.

[7] National Security Agency, *U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness*, Version 0.621, 1 July 2004.

[8] Nguyen, T. D., Levin, T. E., and Irvine, C. E., "TCX Project: High Assurance for Secure Embedded Systems", SIGBED Review, Volume 2, Number 2, April 2005, Special Issue on IEEE RTAS 2005 Work-in-Progress, http://www.cs.virginia.edu/sigbed/vol2_num2.html

[9] Shapiro, J. "EROS: The Extremely Reliable Operating System," 1999. http://www.eros-os.org

[10] "The Fiasco Microkernel," February 2004. http://os.inf.tu-dresden.de/fiasco/

[11] "The Apache Software Foundation," 2005. http://www.apache.org

[12] "OpenSSL," June 16, 2005. http://www.openssl.org/

[13] "Linux Kernel Archives," June 16, 2005. http://www.kernel.org/

[14] National Security Agency, *U.S. Government Web Server Protection Profile for Basic Robustness Environments*, Version 0.41, 1 August 2003.

[15] Kane, D. R., "Web-based dissemination system for the Trusted Computing Exemplar Project," Masters Thesis, Naval Postgraduate School, Monterey, California, June 2005.

[16] National Security Agency, *Consistency Instruction Manual For development of US Government Protection Profiles For use in Medium Robustness Environments*, Release 3.0, 1 February 2005.

[17] Common Criteria Project Sponsoring Organizations, *Common Criteria for Information Technology Security Evaluation*, CCIMB-2005-07-001, Version 3.0 Revision 2, June 2005. http://www.commoncriteriaportal.org